UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/561,941	02/15/2006	Takeshi Inuo	029471-0194	3034

22428 7590 08/14/2007

FOLEY AND LARDNER LLP
SUITE 500
3000 K STREET NW
WASHINGTON, DC 20007

EXAMINER	
VICARY, KEITH E	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
08/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/561,941	INUO, TAKESHI	
	Examiner	Art Unit	
	Keith Vicary	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 May 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18 and 20-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18 and 20-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *5/7/2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-18 and 20-22 are pending in this office action and presented for examination. Claims 1-12, 14-18, and 20-22 are amended and claim 19 is cancelled by amendment filed 5/7/2007.

Specification

2. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The newly amended title which merely recites reconfigurable hardware in various forms is not specific enough to be clearly indicative of the invention to which the claims are detected.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claim 18 is rejected under 35 U.S.C. 102(b) as being anticipated by Smith et al. (Smith) (US PAT 6658564).

5. **Consider claim 18**, Smith discloses a control flow analysis procedure in which the control flow of an application program is analyzed (col. 2, lines 18-20, col. 10, lines 51-53; software development tools; col. 11, lines 1-3; system design language profiler), the application program is divided into processing units (col. 2, lines 1-8), and a

command sequence intermediate code combining commands controlled by reconfigurable hardware that executes the divided processing units within an electronic computer is generated (col. 2, lines 22-26, software and hardware functions; col. 10, lines 60-61); a command sequence implementation procedure in which a command sequences is generated by translating the command sequence intermediate code into a form that can be executed by the electronic computer (col. 2, lines 25-31, col. 11, lines 56-62; threads and configuration data); and a program data generation procedure in which the operational content of a processing unit is translated into a form that can be executed by the electronic computer (col. 12, lines 1-6, linker). Smith also discloses the application program is divided so that each processing unit can be stored in a program data memory that holds a program creating a logic of said reconfigurable (col. 2, lines 1-8, configuration data) when the control flow of the application program is analyzed and divided into processing units in said control flow analysis procedure (col. 2, lines 18-20, col. 8, lines 50-53 and 58-61; col. 11, lines 1-3; system design language profiler).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1, 12, 14, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside et al. (Fallside) (US PAT 6326806) in view of Smith.

8. **Consider claim 1**, Fallside discloses an electronic computer comprising; a processing device (Figure 1 as a whole, specifically including the FPGA 104 and the configuration control circuit 106) including reconfigurable hardware that can create a logic circuit with a program (Figure 1, FPGA 104), and a control device (Figure 1, configuration control circuit 106) executing a command specified by the processing device (col. 4, lines 27-29, initiate reconfiguration), wherein said command is instructed to be executed when the processing device detects a predetermined condition (col. 4, lines 27-29; initiate reconfiguration) and includes a command for execution of switching programs logically creating the reconfigurable hardware (col. 4, lines 27-29, 34-35, signals; col. 6, lines 65-67, instruction signals, analogous to the command).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during runtime, the virtual logic manager may control the swapping of programmed logic

configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application.

9. **Consider claim 12**, Fallside discloses a processing device (Figure 1 as a whole, specifically including the FPGA 104 and the configuration control circuit 106) including reconfigurable hardware that can create a logic circuit with a program (Figure 1, FPGA 104), and a control device (Figure 1, configuration control circuit 106) executing a command specified by the processing device (col. 4, lines 27-29, initiate reconfiguration); wherein said command is instructed to be executed when the processing device detects a predetermined condition (col. 4, lines 27-29, initiate

reconfiguration) and includes a command for execution of switching programs logically creating the reconfigurable hardware (col. 4, lines 27-29, 34-35, signals; col. 6, lines 65-67, instruction signals, analogous to the command); and said processing device comprises a second processing device including reconfigurable hardware that can create a logic circuit with a program (Figure 5, FPGA1-2) and a second control device executing a command specified by the second processing device (Figure 1, configuration control circuit 106).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during runtime, the virtual logic manager may control the swapping of programmed logic configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application.

10. **Consider claim 14**, Fallside discloses issuing an instruction to execute a command (col. 4, lines 27-29, initiate reconfiguration) when a processing device including reconfigurable hardware that can create a logic circuit with a program (col. 4, lines 27-29, 34-35, signals; col. 6, lines 65-67, instruction signals, analogous to the command) detects a predetermined condition (col. 4, lines 27-29, initiate reconfiguration); and executing switching programs that logically create reconfigurable hardware (col. 4, lines 27-29, 34-35; col. 6, lines 65-67) by a control device that has received the command execution instruction from the processing device (Figure 1, configuration control circuit 106).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a

plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during runtime, the virtual logic manager may control the swapping of programmed logic configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in

order to enable optimized execution times and parallelism for a computer handling a given application.

11. **Consider claim 20**, Fallside discloses a procedure in which, when a processing device (Figure 1 as a whole, specifically including the FPGA 104 and the configuration control circuit 106) including reconfigurable hardware that can create a logic circuit with a program (Figure 1, FPGA 104) detects a predetermined condition and issues an instruction to execute a command (col. 4, lines 27-29, initiate reconfiguration), a control device (Figure 1, configuration control circuit 106) that has received the command execution instruction from the processing device executes switching programs logically creating the reconfigurable hardware (col. 4, lines 27-29, 34-35, signals; col. 6, lines 65-67, instruction signals, analogous to the command).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during

runtime, the virtual logic manager may control the swapping of programmed logic configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application.

12. Claims 2 and 4-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside and Smith as applied to claim 1 above, and further in view of Abramovici and Trimberger (US PAT 6573748).

13. Abramovici is cited by the applicant in IDS paper filed 12/22/2005.

14. **Consider claim 2**, Fallside discloses said processing device comprises a plurality of banks (Figure 5, plurality of FPGAs; also col. 8, lines 61-64) each having a

processing element with reconfigurable hardware (Figure 5, plurality of FPGAs; also col. 8, lines 61-64), program data memory holding a program that creates a logic circuit in said reconfigurable hardware (RAM 208 in Figure 5; col. 4, lines 18-20) and connecting it to the outside (Figure 5, data bus).

However, Fallside does not disclose each of a plurality of banks having at least one program data memory each holding a program that creates a logic circuit in said reconfigurable hardware. Furthermore, although Fallside's plurality of FPGAs are shown to be connected to the outside by a data bus, he nevertheless does not explicitly disclose an effective bank selection unit selecting one bank from the plurality of banks, making it effective.

On the other hand, Abramovici does disclose each of a plurality of banks (FPGA 1-4 in Figure 2; col. 4, lines 5-7, FPGAs) having at least one program data memory each holding a program that creates a logic circuit in said reconfigurable hardware (col. 4, lines 14-16 and 57-59; dedicated RAM in the reconfigurable hardware).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that having a separate program data memory for each bank would generally minimize execution time as simultaneous reads and writes by different banks would be possible without increased hardware complexity on the memory itself. Furthermore, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that memory is frequently embedded into reconfigurable hardware to provide temporary data storage.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Abramovici with the invention of Fallside in order to minimize execution time without increasing memory hardware complexity while providing temporary data storage to the reconfigurable hardware.

However, Fallside, Smith, and Abramovici do not disclose an effective bank selection unit selecting one bank from the plurality of banks, making it effective.

Although the use of an effective output selection unit selecting one output from a plurality of outputs, making it effective is well known in the art, Trimberger nevertheless discloses an effective bank selection unit selecting one bank from the plurality of banks, making it effective and connecting it to the outside (Figure 9, 920, 925, 930; col. 6, lines 10-15; the configuration memory is analogous to the bank as it is the data outputs that are relevant).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that using a demultiplexor to select between outputs of anything is well known in the art as a simple way of choosing between two outputs, and using the demultiplexor of Trimberger rather than the data bus of Fallside would result in decreased hardware complexity.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Trimberger with the invention of Fallside, Smith, and Abramovici in order to successfully enable the ability of a plurality of banks to output processed data in the same direction without excess hardware complexity.

15. **Consider claim 4**, Fallside discloses that at least one processing element of said processing device is comprised of reconfigurable hardware and the other processing elements are each comprised of reconfigurable hardware or a general-purpose CPU (Figure 5, plurality of FPGAs; also col. 8, lines 61-64).

16. **Consider claim 5**, Fallside discloses that said control device interprets and executes (col. 6, lines 18-19, signals configuring FGPA); an activate command specifying said effective bank in case where there is a plurality of said banks, and specifying said effective program data memory and activating operation of said specified processing element when there is a plurality of said program data memories (col. 6, lines 46-47, fpga_cs); a halt command halting operation of said specified processing device (col. 6, lines 39-41; the command holds off the configuration operation); an interrupt command issuing an interrupt vector from said control device to said specified processing device (col. 6, lines 36-38; the signal triggers the start-up sequence which is analgous to the interrupt vector in the instant application; alternatively, lines 39-41 for an interrupt in general); a load_prg command transferring program data from a specified memory device to said program data memory (col. 6, lines 42-43, fpga_write); a cancel_prg command canceling the load_prg instruction (col. 6, lines 33-35, fpga_prog), and a wait_prg command waiting until completion of the load_prg instruction (col. 6, lines 44-45; the busy signal being asserted is analogous to the wait command).

17. Claims 3, 16-17, and 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside and Smith as applied to claim 1 above, and further in view of Trimberger.

18. **Consider claim 3**, Fallside discloses that said processing device comprises a bank including a processing element that includes reconfigurable hardware (Figure 1, FPGA 104), a plurality of program data memories each holding a program that creates a logic circuit in said reconfigurable hardware (RAM 208 in Figure 5; col. 4, lines 18-20 and 47-49).

However, Fallside and Smith do not disclose an effective block selection unit selecting one memory from the plurality of program data memories and making it effective.

Although the use of an effective block selection unit selecting one memory from a plurality of memories and making it effective is well known in the art, Trimberger nevertheless discloses an effective block selection unit selecting one memory from the plurality of program data memories and making it effective (Figure 9, 920, 925, 930, memories and muxes; col. 6, lines 10-15, first and second memory spaces).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that using a demultiplexor to select between outputs of memory is well known in the art as a simple way of choosing between two outputs, and using two smaller memories with a demultiplexor can be more economical than using a bigger memory.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Trimberger with the invention of Fallside and Smith in order to successfully enable the ability of two memories without excess hardware.

19. **Consider claim 16**, Fallside discloses issuing an instruction to execute a command (col. 4, lines 27-29, 34-35; col. 6, lines 65-67) when a processing device detects a predetermined condition (col. 4, lines 27-29) said processing device including reconfigurable hardware (Figure 1, FPGA 104), a plurality of program data memories that hold programs creating logic circuits of the reconfigurable hardware (RAM 208 in Figure 5; col. 4, lines 18-20 and 47-49); executing, by a control device that has received the command execution instruction from the processing device an activate command (col. 6, lines 46-47).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into

blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during runtime, the virtual logic manager may control the swapping of programmed logic configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application.

However, Fallside and Smith do not explicitly disclose an effective block selection unit that selects one program data memory from the plurality of program data memories and that makes it effective, and hence does not explicitly disclose controlling the effective block selection unit so as to make a specified program data memory effective and connecting it to the reconfigurable hardware; and switching the content of a logic circuit executed by the reconfigurable hardware.

Although the use of an effective block selection unit selecting one memory from a plurality of memories and making it effective is well known in the art, Trimberger nevertheless discloses an effective block selection unit that selects one program data memory from the plurality of program data memories and that makes it effective, and controlling the effective block selection unit so as to make a specified program data memory effective and connecting it to the reconfigurable hardware; and switching the content of a logic circuit executed by the reconfigurable hardware (Figure 9, 920, 925, 930; col. 6, lines 10-15, memory spaces).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that using a demultiplexor to select between outputs of memory is well known in the art as a simple way of choosing between two outputs, and using two smaller memories with a demultiplexor can be more economical than using a bigger memory.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Trimberger with the invention of Fallside and Smith in order to successfully enable the ability of two memories without excess hardware.

20. **Consider claim 17**, Fallside discloses said control device executes (col. 6, lines 18-19); a halt command halting the operation of said specified processing device (col. 6, lines 39-41; the command holds off the configuration operation); an interrupt command issuing an interrupt vector from said control device to said specified processing device

(col. 6, lines 36-38; the signal triggers the start-up sequence which is analgous to the interrupt vector in the instant application; alternatively, lines 39-41 for an interrupt in general); a load_prg command transferring program data from a specified memory device to said program data memory (col. 6, lines 42-43); a cancel_prg command canceling the load_prg instruction (col. 6, lines 33-35), and a wait_prg command waiting until the completion of the load_prg instruction (col. 6, lines 44-45; the busy signal being asserted is analogous to the wait command).

21. **Consider claim 21**, Fallside discloses a procedure in which, when a processing device including reconfigurable hardware (Figure 1 as a whole, specifically including the FPGA 104 and the configuration control circuit 106), a plurality of program data memories that hold programs creating logic circuits of the reconfigurable hardware (RAM 208 in Figure 5; col. 4, lines 18-20 and 47-49) detects a predetermined condition (col. 4, lines 27-29) and issues an instruction to execute a command (col. 4, lines 27-29, 34-35; col. 6, lines 65-67), a control device (Figure 1, configuration control circuit 106) that has received the command execution instruction from the processing device executes an activate command (col. 6, lines 46-47).

However, although Fallside discloses of multiple logic circuits (Fallside, Figure 5), Fallside does not explicitly disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit. This is because Fallside's invention is directed toward *how* the FPGAs are configured but not *what* they are configured with.

On the other hand, Smith does disclose of a device for dividing an application program into a plurality of processing units, wherein the aforementioned logic circuit is created for each said processing unit (col. 2, lines 1-8, when an application is compiled, the functions of the application that are implemented in hardware are partitioned into blocks containing configuration data; col. 2, lines 66-67 and col. 3, lines 1-6, during runtime, the virtual logic manager may control the swapping of programmed logic configuration data and application state information between programmable logic resources and a secondary storage device).

The teaching of Smith allows for optimized execution times and parallelism for a computer handling a given application (Smith, col. 1, lines 44-47 and lines 56-60).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the environment of Smith is analogous to the environment of Fallside as smith also contains logic circuits that are dynamically configurable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Smith with the invention of Fallside in order to enable optimized execution times and parallelism for a computer handling a given application.

However, Fallside does not explicitly disclose an effective block selection unit that selects one program data memory from the plurality of program data memories and

that makes it effective, and hence does not explicitly disclose controlling the effective block selection unit so as to make the specified program data memory effective and switch connection to the reconfigurable hardware.

Although the use of an effective block selection unit selecting one memory from a plurality of memories and making it effective is well known in the art, Trimberger nevertheless discloses an effective block selection unit that selects one program data memory from the plurality of program data memories and that makes it effective, and controlling the effective block selection unit so as to make a specified program data memory effective and switch connection to the reconfigurable hardware (Figure 9, 920, 925, 930; col. 6, lines 10-15).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that using a demultiplexor to select between outputs of memory is well known in the art as a simple way of choosing between two outputs, and using two smaller memories with a demultiplexor can be more economical than using a bigger memory.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Trimberger with the invention of Fallside and Smith in order to successfully enable the ability of two memories without excess hardware.

22. **Consider claim 22**, Fallside discloses a procedure in which a halt command halting the operation of said specified processing device (col. 6, lines 39-41; the

command holds off the configuration operation); an interrupt command issuing an interrupt vector from said control device to said specified processing device (col. 6, lines 36-38; the signal triggers the start-up sequence which is analgous to the interrupt vector in the instant application; alternatively, lines 39-41 for an interrupt in general); a load_prg command transferring program data from a specified memory device to said program data memory (col. 6, lines 42-43); a cancel_prg command canceling the load_prg instruction (col. 6, lines 33-35), and a wait_prg command waiting until the completion of the load_prg instruction (col. 6, lines 44-45; the busy signal being asserted is analogous to the wait command).

23. Claims 6-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside and Smith as applied to claim 1 above, and further in view of Birns et al. (Birns) (US PAT 5887189).

24. **Consider claim 6**, although Fallside discloses reading commands, interpreting, and executing it (col. 9, lines 12-20; the FPGA provides the desired configuration instruction *signals* to configuration control circuit and then triggers the reconfiguration, with the signals being CFG_MODE in col. 7, lines 3-5; also note that he also discloses that the configuration control circuit could be implemented as a microcontroller, col. 4, lines 56-57); Fallside nevertheless does not disclose a command code memory holding commands that said control device executes, wherein said control device comprises a command code reference device reading commands from the command code memory

according to an address specified by said processing device, interpreting, and executing it.

On the other hand, Birns does disclose a command code memory holding commands (Fig. 1, instruction memory 18) that said control device executes (col. 9, lines 49-51) wherein said control device comprises a command code reference device reading commands from the command code memory (col. 3, lines 27-31; decode unit) according to an address specified by said processing device (col. 9, lines 55-57), interpreting, and executing it (col. 9, lines 49-51).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that a control device such as a microsequencer that has instructions stored in memory that can be initiated when given an address is more configurable and cost effective than a control device which executes commands based on predefined signals and not addresses, as to compensate, each of the external devices issuing said signals would need to have additional hardware to implement a series of instructions. Furthermore, Fallside discloses the potential use of a microsequencer as a control device as noted above (col. 4, lines 56-57).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Birns with the invention of Fallside and Smith in order to allow greater configurability and cost-effectiveness.

25. **Consider claim 7**, the claim is rejected for same reasons as claim 6 above. Furthermore, Fallside and Birns discloses that said command code reference device

comprises an address counter holding the address of said command code memory (Birns, col. 9, line 56, program counter), and in the exchange of commands between said processing device and said control device (Fallside, col. 6, lines 18-19), a first address control line indicating that an address signal line outputted by said processing device is effective (Fallside, col. 6, line 17, output-enable signals; col. 7, lines 5-7; with the Mode Enable analogous to the Address enable), and a second address counter control line instructing whether the value of the address signal line is stored in the address counter as it is (Birns, col. 9, lines 66-67 and col. 10, line 1; absolute addresses) or the result of adding the value of the address signal line to the value of the address counter is stored in the address counter when the first control line is effective (Birns, col. 9, lines 55-57; relative branches and displacement).

26. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside, Smith, and Birns as applied to claim 7 above, and further in view of Stewart et al. (Stewart) (US PAT 5473763).

27. **Consider claim 8**, Birns discloses said commands are stored in said command code memory in a format comprising a command code that classifies the commands (col. 3, lines 29-32; because the instructions are decoded, it is inherent that they are represented as some form of opcode), an address counter control code (col. 9, lines 66-67 and 55-57), and said address counter control code includes a load adr command setting the value of the address counter (col. 9, lines 66-67) and a add_adr command adding a specified value to the address counter (col. 9, lines 55-57).

However, Birns does not explicitly disclose a flag that indicates whether or not the following command is executed.

On the other hand, Stewart does disclose a flag that indicates whether or not the following command is executed.

The use of a flag that indicates whether a following command is executed is a common way of putting a processor or microcontroller into idle mode (col. 7, lines 22-25) that doesn't require the use of repeated nop instructions, which typically lowers power consumption. Furthermore, the disclosed stop bit of Stewart fits into the environment of Fallside and Birns as the invention of Stewart deals with running certain program sequences at a starting address (col. 3, lines 26-32) upon the activation of an external interrupt trigger (col. 3, lines 38-40), which is analogous to Fallside and Birns running certain program sequences upon the receipt of an address and enabling signal from an external reconfigurable logic.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Stewart with the invention of Fallside, smith, and Burns in order to save power.

28. **Consider claim 9**, Birns discloses said address counter control code includes a push_adr command that hides the address counter in an address counter stack provided in said control device and that sets a new value to the address counter, and a pop_adr command that returns the value of the address counter stack to the address

counter (both of these commands are inherent in col. 10, lines 2-3, return address stack).

29. Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside and Smith as applied to claim 1 above, and further in view of Sachs et al. (Sachs) (US PAT 4860192).

30. **Consider claim 10**, Fallside and Smith do not disclose a cache device including a cache memory that temporarily holds data to be transferred to said processing device and a cache controller that controls the cache memory wherein the cache controller is controlled by a command issued by said processing device.

On the other hand, Sachs does disclose a cache device including a cache memory (col. 1, line 18, cache memory) that temporarily holds data to be transferred to said processing device (col. 1, lines 37-41, cache memory) and a cache controller that controls the cache memory wherein the cache controller is controlled by a command issued by said processing device (col. 1, line 18, cache controller).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that Implementing a cache in general allows for faster memory accesses, leading to accelerated data transfer and reduced execution time. Furthermore, the use of a cache specifically for holding configurations can allow for specialized direct output to the reconfigurable hardware, facilitating wide parallel loading of the configuration data and reducing configuration times.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Sachs with the invention of Fallside and Smith to reduce total execution and configuration time.

31. **Consider claim 11**, the claim is rejected for same reasons as claim 10 above. Furthermore, Sachs said cache device comprises an address translation device that translates an address defined externally to said processing device into an address defined inside of the processing device, and the address translation device is controlled by a command issued by said processing device (col. 1, lines 19, 32-40; the externally defined address is the main memory, the internally defined address is the cache).

32. Claims 13 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fallside and Smith as applied to claims 1 and 14 above, and further in view of Abramovici.

33. **Consider claim 13**, Fallside and Smith do not disclose a semiconductor integrated circuit implementing the electronic computer as defined in claim 1.

On the other hand, Abramovici does disclose a semiconductor integrated circuit implementing the electronic computer as defined in claim 1 (col. 4, lines 39-40)

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that Implementing an electronic computer on a semiconductor integrated circuit is an optimal method of doing so for both space and performance considerations.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Abramovici with the invention of Fallside and Smith because of space and performance considerations.

34. **Consider claim 15**, Fallside and Smith do not disclose, after said switching, while a program in a predetermined program data memory is being executed, a next program is read into another program data memory.

On the other hand, Abramovici does disclose, after said switching, while a program in a predetermined program data memory is being executed, a next program is read into another program data memory (col. 2, lines 18-21,col. 5, lines 40-41, 53-58; note that the loading of a page into memory is based solely on the input buffer for an unloaded page becoming full, and there is nothing that would suggest that this loading of a page into memory would need to wait until other programs in another memory are finished executing).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that loading a program into a program data memory while another program in a different program data memory is a well-known method for reducing stalls and the total execution time for a processor, and is easily applicable to the environment of the invention of Fallside.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Abramovici with the invention of Fallside and Smith in order to reduce the total amount of execution time for a processor.

Response to Arguments

35. Applicant argues on page 13 that Fallside does not disclose the using a FPGA as reconfigurable hardware. However, a cited above, Fallside does disclose this in, for example, col. 4, lines 27-29, the FPGA is also configured to initiate reconfiguration. Applicant argues that Fallside fails to disclose a device and method for dividing an application program into a plurality of processing units. Smith has been brought into the majority of the independent claims which recite this limitation to teach this limitation. Applicant argues that while Fallside teaches a control circuit for reconfiguring a FPGA in response to a predetermined condition, Fallside does not teach a method and a processing device including reconfigurable hardware that can create a logic circuit for each processing unit. However, as recited above, Fallside does indeed teach creating a logic circuit; and Smith teaches that the logic circuit is created for each processing unit. Applicant argues that in contrast to the claimed invention in claims 1 and 5, the instruction signals of Fallside exist only in the FPGA 0. However, applicant also cites Figure 5, which shows the instruction signals traveling between configuration control circuit and FPGA 0. Moreover, Fallside discloses that there are numerous alternative reconfiguration arrangements for multiple FPGAs and gives various examples in col. 9, lines 1-6.

36. Applicant argues that while Smith mentions that functions of an application are partitioned into blocks, Smith does not disclose a method that can create a logic circuit for each processing unit because Smith only executes partitioning at a functional level.

However, as cited above, Smith does disclose creating a logic circuit for each processing unit in, for example, col. 3, lines 1-5, swapping of programmable logic configuration data; the programmable logic configuration data controls the creation of the logic circuit, or col. 8, lines 50-53, allocating programmable logic resources to functions). Although the specifics as to how Smith accomplishes this are not disclosed as the majority of Smith's invention is directed toward the functional level, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that Smith's teaching would be able to be implemented into the invention of Fallside, and it would have been obvious to one of ordinary skill in the art at the time of the invention that Smith's teaching of dividing a program into functional units *in order to implement in a programmable logic device* would entail the functional units being executed on different hardware configurations as otherwise there would be no reason to divide the program.

37. Applicant argues on page 14 that the motivation for applicant's invention differs from the motivation given in the rejection. However, the motivation given in the rejection is still valid and moreover, a plurality of the banks is not inherently necessary in order to switch the divided application, as multiple configurations could be stored in the same bank. Applicant argues that FPGA_PROG operates analogous to the claimed cancel_prg; examiner does not disagree, however if this was a typographical error, FPGA_PROG as disclosed by Fallside does operate analogously to the cancel_prg operation as claimed. The cancel_prg operation as claimed merely cancels the loading instructions; and Fallside's FPGA_PROG operation resets the internal configuration

logic, which cancels the effects of the load operation. The two commands are analogous given a broad interpretation. Moreover, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that specific commands at this level of the invention would be a design choice given that the functionality is analogous to that claimed.

Conclusion

38. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Greenberg (US 20030056091) discloses of scheduling in a reconfigurable hardware architecture with multiple hardware configurations.

b. Morelli et al. (US 6438737) discloses of reconfigurable logic with memory storing logic designs and program division.

c. Greenbaum et al. (US 6077315) discloses of reconfigurable computing and program division similar to the arts above.

39. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Keith Vicary whose telephone number is (571) 270-1314. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

kv

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100